Bounds for the Query Complexity of Approximate Equilibria

PAUL W GOLDBERG, University of Oxford
AARON ROTH, University of Pennsylvania

We analyze the number of payoff queries needed to compute approximate equilibria of multi-player games. We find that query complexity is an effective tool for distinguishing the computational difficulty of alternative solution concepts, and we develop new techniques for upper- and lower bounding the query complexity. For binary-choice games, we show logarithmic upper and lower bounds on the query complexity of approximate correlated equilibrium. For *well-supported* approximate correlated equilibrium (a restriction where a player's behavior must always be approximately optimal, in the worst case over draws from the distribution) we show a linear lower bound, thus separating the query complexity of well supported approximate correlated equilibrium from the standard notion of approximate correlated equilibrium.

Finally, we give a query-efficient reduction from the problem of *computing* an approximate well-supported Nash equilibrium to the problem of verifying a well supported Nash equilibrium, where the additional query overhead is proportional to the description length of the game. This gives a polynomial-query algorithm for computing well supported approximate Nash equilibria (and hence correlated equilibria) in concisely represented games. We identify a class of games (which includes congestion games) in which the reduction can be made not only query efficient, but also computationally efficient.

Categories and Subject Descriptors: F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; J.4 [**Social and Behavioral Sciences**]: Economics

Additional Key Words and Phrases: Payoff queries, correlated equilibrium

1. PRELIMINARIES

This paper compares the *query complexity* of alternative game-theoretic solution concepts. Instead of a game G being presented in its entirety as input to an algorithm $\mathcal{A}$, we assume that $\mathcal{A}$ may submit queries consisting of strategy profiles, and get told the resulting payoffs to the players in G. This model is appealing when G has many players, in which case a naive representation of G would be exponentially-large. Assuming G belongs to a known class of games $\mathcal{G}$, this gives rise to the question of how many queries are needed to find a solution, such as exact/approximate Nash/correlated equilibrium. One can study this question is conjunction with other notions of cost, such as runtime of the algorithm. An appealing aspect of query complexity is that it allows new upper and lower bounds to be obtained, providing a mathematical criterion to distinguish the difficulty of alternative solution concepts, as discussed in more detail below.

We consider queries that consist of pure-strategy profiles, and in which the answer to any query is the payoffs that all the players receive from that profile. In this paper

This work is supported in part by EPSRC under grant EP/K01000X/1, an NSF CAREER award, under NSF grants CCF-1101389 and CNS-1065060, and a Google Focused Research Award.
Authors' addresses: P.W. Goldberg, Department of Computer Science, Wolfson Building, Parks Road, Oxford OX1 3QD, U.K.; A. Roth, Department of Computer and Information Science, Levine Hall, 3330 Walnut Street, Philadelphia, PA 19104-6389, USA.

EC'14, June 8–12, 2014, Stanford, CA, USA. ACM 978-1-4503-2565-3/14/06 ...$15.00.

http://dx.doi.org/10.1145/2600057.2602845

we mostly focus on n-player binary-action games, which have 2^n pure profiles. In the context of approximate equilibria, we use ϵ to denote the bound on a player's incentive to deviate, and we make the standard assumption that all payoffs lie in the range $[0, 1]$. We are interested in algorithms with query complexity at most polynomial in n, meaning of course that only a very small fraction of a game's profiles may be queried.

Notation. We have n players denoted by the numbers $\{1, 2, \ldots, n\}$. Let A^i be the set of possible actions, or pure strategies, of player i and let $A = A^1 \times \ldots \times A^n$ be the set of pure profiles. In this paper we assume all players have the same number m of pure strategies, i.e. $|A^i| = m$ for all i. $u^i : A \longrightarrow [0, 1]$ denotes player i's utility function. Generally $\mathcal{G}$ will denote a class of games, and G denotes a specific game. $\mathcal{G}_n$ denotes n-player *binary-choice* games, in which $m = 2$.

1.1. Alternative definitions of approximate equilibrium

We review the notions of exact and approximate correlated equilibrium (CE), and introduce the definition of *well-supported* approximate CE. A probability distribution ψ on A is a *correlated equilibrium* if for every player i, all pure strategies $j, k \in A^i$ we have (letting (k, a^{-i}) denote the profile a with i's strategy replaced with k)

$$\sum_{a \in A : a^i = j} \psi(a)[u^i(k, a^{-i}) - u^i(a)] \leq 0. \tag{1}$$

An *ϵ-approximate correlated equilibrium* (ϵ-CE) is a distribution ψ where for every player i, every function $f : A^i \longrightarrow A^i$, we have

$$\sum_{a \in A} \psi(a)[u^i(f(a^i), a^{-i}) - u^i(a)] \leq \epsilon. \tag{2}$$

The above definition is based on swap regret, from [Blum and Mansour 2007], although other definitions (e.g. based on internal regret) are possible. An alternative definition from [Hart and Mas-Colell 2000] of *correlated ϵ-equilibrium* replaces the RHS of (1) with $\epsilon > 0$. The definitions are not quite the same: an ϵ-CE is a correlated ϵ-equilibrium, while a correlated ϵ-equilibrium need only be a $m\epsilon$-CE.

An *ϵ-approximate coarse correlated equilibrium* (ϵ-CCE) is a distribution ψ in which for all players i, strategies j,

$$\sum_{a \in A} \psi(a)[u^i(j, a^{-i}) - u^i(a)] \leq \epsilon.$$

In general an ϵ-CE is an ϵ-CCE, but the converse does not hold.[1] However, in the case of binary-choice games (the class of games we mainly consider here) an ϵ-CE is an ϵ-CCE, while an ϵ-CCE is a 2ϵ-CE, hence the two notions are basically the same from the perspective of our interest in asympototic query complexity in terms of n and ϵ.

An *ϵ-well-supported approximate correlated equilibrium* (ϵ-WSCE) imposes the more demanding requirement that *after* a player observes his own action, his expected gain from switching to any other action is at most ϵ. It can be precisely defined by saying that for any player i, strategies $j, k \in A^i$, letting $p^i_j = \Pr_{a \sim \psi}[a^i = j]$,

$$\sum_{a \in A : a^i = j} \psi(a) u^i(k, a^{-i}) - \sum_{a \in A : a^i = j} \psi(a) u^i(j, a^{-i}) \leq p^i_j \epsilon$$

[1] For example, in the game of rock-paper-scissors, the uniform distribution over the 3 strategy profiles in which both players play the same strategy is a CCE but not a CE.

which is equivalent to $\mathbb{E}[u^i(k, a^{-i})] - \mathbb{E}[u^i(j, a^{-i})] \leq \epsilon$, where the expectations are w.r.t. ψ restricted to profiles where i plays j.

It may be helpful to note the following comparison with the earlier definition of correlated ϵ-equilibrium. Observe that if $\sum_{a \in A: a^i = j} \psi(a)$ is small (meaning that it is unlikely that in a random profile, player i plays j) then putting some given $\epsilon > 0$ into the RHS of (1) introduces more slackness than would be the case if $\sum_{a \in A: a^i = j} \psi(a)$ were large. The definition of ϵ-WSCE corresponds to an attempt to redress this variable slackness.

OBSERVATION 1. *Let G be a game and fix $\epsilon > 0$. The set of ϵ-WSCE of G is convex.*

PROOF. Let ψ and ψ' be two ϵ-WSCE of G. We show that $\psi'' = \lambda\psi + (1 - \lambda)\psi'$ is also an ϵ-WSCE (for $\lambda \in (0, 1)$). Suppose strategy profile s is sampled from ψ'' and some player i observes his action a, i.e. his marginal observation of s on his own behavior. If there was some action a' that would pay i more then ϵ more (in expectation) then one (or both) of ψ or ψ' would have to have that property. □

For completeness, recall that ψ is a *Nash equilibrium* (NE) if it obeys the further constraint that ψ is a product distribution of its restriction to the individual players.

Example 1.1. In the *directed path graphical game* G_n, each player $i \in \{1, 2, \ldots, n\}$ has two actions, 0 and 1. Player 1 gets paid 1 for playing 1 and 0 for playing 0. For $i > 1$, player i gets paid 1 for copying player $i - 1$ and 0 for playing the opposite action.

Observations about Example 1.1. In an exact Nash equilibrium of G_n, all players play 1 with probability 1. Furthermore it is not hard to see that for $\epsilon < 1$, the only ϵ-WSCE requires all players to play 1 with probability 1. In an ϵ-Nash equilibrium of G_n, for small ϵ all players play 1 with high probability. For example, putting $\epsilon = \frac{1}{100}$, it can be proved by induction on i that player i plays 1 with probability $> \frac{9}{10}$. By contrast, for any $\epsilon > 0$ there exist ϵ-CE where the probability that i plays 1 can decrease towards 0 as i increases. Specifically, let $z \sim U[0, 1]$; if $z \in [r\epsilon, (r + 1)\epsilon]$ let players $\{1, \ldots, r\}$ play 1 and let the other players play 0. It can be checked that the resulting distribution over pure-strategy profiles is an ϵ-approximate CE.

These observations indicate that for some games, there are many approximate correlated equilibria that are ruled out when the "well-supported" requirement is imposed.

1.2. Related work

Various game-theoretic settings have been studied in which an agent's type may be exponentially-complex (raising the spectre of exponential communication complexity), but a solution may be found via a feasibly small sequence of queries to an agent about his type. Some examples are the following. In the setting of combinatorial auctions where buyers have valuations for bundles (subsets) of a set of items, [Conen and Sandholm 2001] present algorithms that exploit properties of valuation functions in order to find a good allocation without querying the functions exhaustively. [Blum et al. 2004] study (from the learning-theory perspective) welfare-optimal allocation of items amongst buyers, obtained via *value queries* in which the query elicits the value a buyer places on a bundle. [Conitzer 2009] studies rank aggregation in a setting where a voter may be asked which of 2 alternatives he prefers; that paper makes the point that to find a winner, it is not always necessary to learn all voters' complete rankings (although, the query complexity of finding an aggregate ranking is similar to that of learning all the voters' rankings).

Work on the query complexity of the solution concepts of Section 1.1 for general multi-player games is more recent. [Fearnley et al. 2013] point out that payoff queries

have a motivation coming from empirical game-theoretical analysis; some alternative queries are mentioned. They present algorithms for congestion games and bimatrix games. For general multiplayer games (the subject of this paper), payoff query complexity has recently been studied by [Hart and Nisan 2013; Babichenko and Barman 2013; Babichenko 2013]. The main focus of these papers is on exponential (in n) lower bounds for solutions of n-player games having a constant number m of pure strategies per player. For *deterministic* algorithms, approximate Nash/CE needs exponentially-many queries [Hart and Nisan 2013], which motivates a focus on randomized algorithms (in this paper, all algorithms are randomized). [Hart and Nisan 2013] also shows that *exact* Nash/CE have exponential query complexity even with randomness, which motivates a focus on approximate solution concepts. The main open question (noted in [Hart and Nisan 2013]) is the query complexity of ϵ-NE; in this paper we give a positive result[2] for the special case of concisely-represented games (Theorem 3.3). [Hart and Nisan 2013; Babichenko and Barman 2013] note that (with randomization) one can obtain query-efficient algorithms for approximate correlated equilibrium by simulating regret-based algorithms. In this paper we analyze this query complexity in more detail, and use it as a criterion to distinguish the difficulty of approximate CE from well-supported approximate CE. Most recently, [Babichenko 2013] shows that ϵ-well-supported approximate NE needs exponentially-many queries, for $m = 10^4$ and $\epsilon = 10^{-8}$; it is currently an open question whether these constants can be improved.

ϵ-WSCE is a refinement of approximate CE that is analogous to well-supported approximate Nash equilibrium, studied in [Kontogiannis and Spirakis 2010; Fearnley et al. 2012; Goldberg and Pastink 2014; Babichenko 2013]. In an ϵ-Nash equilibrium (ϵ-NE), a player's payoff is allowed to be up to ϵ worse than his best response. The motivation behind the "well-supported" refinement is that in an ϵ-NE, it may still be the case that a player allocates positive probability to some strategy whose payoff is more than ϵ worse than his best response. Such behavior is disallowed in a well-supported ϵ-NE. Under this more demanding definition of approximate Nash equilibrium, the values of ϵ known to be achievable in polynomial time (in the context of bimatrix games) are accordingly higher; ϵ-NE can be computed for ϵ slightly above $\frac{1}{3}$ [Tsaknakis and Spirakis 2008] while for ϵ-WSNE, the best value known is slightly less than $\frac{2}{3}$ [Fearnley et al. 2012]. For the games studied in this paper, we will see that the payoff query complexity of ϵ-WSCE is strictly higher than that of ϵ-CE. The technique of [Daskalakis et al. 2009] (Lemma 4.28) (see also [Chen et al. 2009] Lemma 3.2) for converting approximate NE into approximate WSNE works for Nash equilibria but not for correlated equilibria, and so we need new upper-bounding techniques.

Communication complexity was analysed for n-player binary-choice games by Hart and Mansour [Hart and Mansour 2010], and for bimatrix games in [Goldberg and Pastink 2014]. [Hart and Nisan 2013] notes that payoff-query bounded algorithms can be efficiently converted into communication-bounded algorithms. Lower bounds seem to be easier to obtain in the payoff-query setting. In the communication-bounded setting, [Hart and Mansour 2010] show efficient upper bounds for ϵ-CE, and exponential lower bounds just for exact (pure or mixed) Nash equilibria. The communication complexity of finding an exact correlated equilibrium is polynomial in the number of players, in contrast with the exponential requirement of NE. It uses Papadimitriou's approach [Papadimitriou 2005] (using the Ellipsoid algorithm) to compute a mixture of product distributions that constitutes a CE. The algorithm interacts with the game using mixed-strategy payoff queries and receiving exact answers. Note that mixed-strategy payoff queries can be approximately simulated by randomly-sampled pure-

[2]That is, a query-efficient algorithm, but one that need not be computationally-efficient.

strategy payoff queries; this suggests that pure payoff queries can be used by a randomized algorithm to find approximate correlated equilibria.

The connection between no-regret algorithms and equilibrium notions, including correlated equilibria, were first noted by [Freund and Schapire 1999; Hart and Mas-Colell 2000]. For an excellent survey of this connection, see [Blum and Mansour 2007]. Appendix B.3 of [Hart and Mansour 2010] makes the observation that regret-minimization techniques yield simple polynomial upper bounds for the communication complexity of approximate CE. We note that this straightforward approach also gives a polynomial upper bound on the number of payoff queries needed to compute approximate correlated equilibrium (but not well-supported correlated equilibrium). We improve this straightforward polynomial dependence on n to an $O(\log n)$ dependence, and give a matching lower bound. To our knowledge, we are the first to study bounds on computing well supported correlated equilibria, which do not follow from no-regret guarantees.

1.3. Our results and techniques

Section 2 gives bounds on the query complexity of ϵ-CE in terms of n and ϵ, while Section 3 gives bounds on the query complexity of ϵ-WSCE in terms of n and ϵ. As functions of the number of players n, the bounds for ϵ-CE are logarithmic. For ϵ-WSCE we have a linear lower bound, thus showing a separation between the two solution concepts. The following observation is a useful starting-point:

OBSERVATION 2. *For binary-choice games the uniform distribution is a $\frac{1}{2}$-approximate Nash equilibrium, thus also an ϵ-approximate CE for any $\epsilon \geq \frac{1}{2}$.*

In Section 2 we show that for any $\epsilon < \frac{1}{2}$, the query complexity is $\Theta(\log n)$, showing that the constant $\frac{1}{2}$ in Observation 2 represents a crisp threshold at which the query complexity becomes non-trivial.

For (non-well-supported) correlated equilibria, our algorithms for query-efficient upper bounds use the Multiplicative Weights algorithm, applying it in two alternative ways. To obtain an upper bound that works well for the case of many strategies per player (Section 2.1), a polynomial (in the number of players) query upper bound follows straightforwardly from an application of multiplicative weights: the standard proof is included in the full version of the paper. We note that this also yields an $O(m \log m)$ upper bound for the problem of computing an approximate Nash equilibrium in a 2-player $m \times m$ zero-sum game, which is substantially smaller than the representation size of the game matrix.

The case of many players having just 2 strategies (Section 2.2) is more subtle: a straightforward application of multiplicative weights would yield a superlinear query complexity upper bound (in the number of players n), but as we show, this linear dependence on n is not necessary: we are instead able to achieve a logarithmic upper bound. The reason the naive application of multiplicative weights requires a linear number of queries is because at every round of the algorithm, it is necessary to estimate the payoff for every action of each of the n players. We observe that if we had a guarantee that at each stage of the algorithm, every player was playing a mixed strategy that placed at least $\Omega(\epsilon)$ probability mass on each action, then it would be possible to estimate the payoffs of every player simultaneously from a single sample consisting of $\approx \log(n)/\epsilon^3$ randomly chosen value queries. Moreover, we can enforce this condition by having each player play according to multiplicative weights coupled with a Bregman projection into an appropriately defined convex polytope. We show that even with this projection, play still converges to an approximate correlated equilibrium. Using this technique we obtain quite an efficient upper bound on the query complexity of

computing an ϵ-CE, using $O(\log n/\epsilon^5)$ queries. The lower bound of Section 2.3 shows that $\Omega(\log n)$ queries are indeed required for any $\epsilon < \frac{1}{2}$. Theorem 2.4 contrasts with the exponential lower bound for deterministic algorithms [Babichenko and Barman 2013; Hart and Nisan 2013].

The lower bounds of Theorems 2.8 and 3.1 work by identifying distributions over the payoff functions of the target game, so that Yao's minimax principle can be applied to algorithms having lower query complexity. Finally, we apply Theorem 3.1 to show that small-support approximate CE are harder (in the query complexity sense) to find than are larger-support approximate CE.

Finally, we give query efficient reductions from the problem of computing an ϵ-approximate well supported Nash equilibrium to the problem of verifying an ϵ-approximate well supported Nash equilibrium (and hence correlated equilibrium), where the query overhead is proportional to the description length of the game. Since *verifying* equilibria can be done query efficiently, this gives query efficient algorithms for computing approximate well supported equilibria in concisely represented games. In special classes of games (including some congestion games) in which the payoff function can be represented as a linear function over polynomially many dimensions, this reduction can be made computationally efficient as well. The main technique is to use no-regret algorithms as mistake bounded learning algorithms for the underlying game, which is similar to how no-regret algorithms have been recently used in data privacy [Roth and Roughgarden 2010; Hardt and Rothblum 2010; Gupta et al. 2012]. We use the algorithms to learn a hypothesis game representation: at each stage, we compute an equilibrium of the hypothesis game, and then make polynomially many queries to check if our computed equilibrium is an equilibrium of the real game. If it is, we are done: otherwise, we have forced the learning algorithm to make a mistake, which we charge to its mistake bound.

2. BOUNDS FOR THE QUERY COMPLEXITY OF ϵ-CE

We use the Multiplicative Weights algorithm to get upper bounds on the query complexity of coarse correlated equilibrium. The first one is applied to zero-sum bimatrix games, and the second one is applied to n-player binary-action games.

2.1. Upper bound for ϵ-approximate coarse correlated equilibrium; few players, many strategies

As a warmup, we consider a straightforward upper bound on the query complexity of computing approximate coarse-correlated equilibria that follows from using no-regret algorithms. In the special case of two player zero-sum games, [Freund and Schapire 1999] showed that two no regret algorithms played against each other converge to an approximate Nash equilibrium. We here observe that this approach yields an upper bound for the query complexity of these equilibrium concepts. The proof is standard, and we include the algorithm and the proof in the full version for completeness.

Theorem 2.1 identifies useful bounds in the case of $m \gg n$; in particular it has an interesting application to the case of 2-player zero-sum games in Corollary 2.2.

THEOREM 2.1. *Let G be a game with n players, each with m pure strategies where $m \geq n$; payoffs lie in $[0,1]$. With probability $1 - nm^{-\frac{1}{8}}$, Algorithm* APPROX COARSE CE *(Figure 1) finds an ϵ-approximate coarse correlated equilibrium of G, using $O(\frac{nm \log m}{\epsilon^2})$ payoff queries.*

COROLLARY 2.2. *Let G be a $m \times m$ zero-sum bimatrix game. It is possible to efficiently compute (w.h.p., using a randomized algorithm) an ϵ-Nash equilibrium of G, using $O(\frac{m \log m}{\epsilon^2})$ payoff queries.*

APPROX COARSE CE
Parameters: learning rate $\eta = \frac{\epsilon}{3}$; number of iterations $T = \frac{9 \ln m}{\epsilon^2}$.
$w_j^i(t)$ is the weight of the j-th strategy of player i at iteration t;
initially $w_j^i(1) = 1$ for all i, j. Do the following for iterations $t = 1, \ldots, T$:

(1) For each player i, let $\mathbf{p}^i(t) = (p_1^i(t), \ldots, p_m^i(t))$ be the probability distribution over i's strategies corresponding to the weights; $p_j^i(t) = w_j^i(t) / \sum_j w_j^i(t)$.

(2) For each player i, sample a pure strategy $s^i(t)$ from $\mathbf{p}^i(t)$. Let $\mathbf{s}(t) = \times_i s^i(t)$ be the resulting strategy profile.

(3) $\forall i$, query the payoffs of all responses to $\mathbf{s}(t)$. Let $m_j^i(t)$ be the cost (negated payoff) of strategy j.

(4) update weights: for each i, j, set $w_j^i(t+1) = w_j^i(t)(1 - \eta m_j^i(t))$.

Output: The distribution $\mathbf{p} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{p}(t)$.

Fig. 1. Using (the naive) application of Multiplicative Weights to compute an approximate coarse CE

This follows from the observation that for zero-sum bimatrix games, an ϵ-approximate coarse correlated equilibrium ψ can be converted to a 2ϵ-NE by taking the product of the marginal distributions of ψ for each player. This result was recently extended to well-supported ϵ-NE [Fearnley and Savani 2013].

2.2. Upper bound for n players, binary actions

We noted at the end of Section 1.2 that no-regret algorithms can be applied directly to yield a polynomial upper bound on the query complexity of ϵ-CE of binary-choice games (more generally for ϵ-CCE when players have more than 2 actions). In this section, we take a more sophisticated approach to show a logarithmic upper bound. We will give a matching lower bound to show that this is optimal.

First, we give the intuition for the approach. The naive application of no-regret algorithms gives a query complexity that is linear in the number of players n because we must take samples to estimate the payoffs of the 2 actions for each of the n players. The improved algorithm in this section is based on the following two optimizations:

(1) If the distribution over game states induced by the mixed strategies of all n players were such that each player had probability mass at least ϵ on each of his two actions, then it would be enough to take $O(\log(n)/\epsilon^3)$ samples from this joint distribution to simultaneously estimate the payoffs of all n players. We formalize this below in Proposition 2.3.

(2) We can simulate play of the game in such a way so as to force each agent to play a mixed strategy that puts weight at least $\epsilon/4$ on each of their two actions. One way to do this is to have each player play multiplicative weights, coupled with a Bregman projection into the polytope $K_{\epsilon/4}$, where $K_{\epsilon/4}$ is the 2-dimensional probability simplex over distributions p, intersected with the linear inequalities constraining $||p||_\infty \geq \epsilon/4$. The result is that the final empirical average of player strategies will always place minimum weight $\epsilon/4$ on any action, and each player will have no regret to the best mixed strategy in $K_{\epsilon/4}$ (see e.g. [Arora et al. 2012]). However, because the total variation distance between *any* probability distribution and $K_{\epsilon/4}$ is at most $\epsilon/2$, this means that no player will have regret greater than $\epsilon/2$ to *any* strategy. Hence, the resulting play will be an $\epsilon/2$-approximate coarse correlated equilibrium, and therefore an ϵ-approximate correlated equilibrium.

The algorithm is given in Figure 2. Using standard concentration bounds we can establish the following fact:

PROPOSITION 2.3. *Let* s *be a mixed-strategy profile of an* n*-player 2-action game, having the property that for any player* i *and strategy* j*,* i *plays* j *with probability at least* γ*. Let* $\mathsf{s}_i(a)$ *be the expected payoff to* i *when* i *plays* a *and the other players play* s_{-i}*.*

With probability $1-\delta$ *we can find, with additive error* $\beta \leq \gamma/2$*, all the* $\mathsf{s}_i(a)$ *values, using* N *payoff queries randomly sampled from* s*, whenever* $N \geq \max\left\{\frac{1}{\gamma\beta^2}\log\left(\frac{8n}{\delta}\right), \frac{8}{\gamma}\log\left(\frac{4n}{\delta}\right)\right\}$.

THEOREM 2.4. *Let* G *be a game with* n *players, each with* 2 *actions; payoffs lie in* $[0,1]$*. For any* ϵ*, with probability* $1-\frac{2}{n}$*, Algorithm* APPROX COARSE CE (2) *(Figure 2) finds an* ϵ*-approximate coarse correlated equilibrium of* G*, using* $\tilde{O}(\frac{\log n}{\epsilon^5})$ *payoff queries.*

PROOF. If Algorithm APPROX COARSE CE (2) updated its probabilities using the true losses of each action $m^i_j(t)$ (rather than with the noisy estimates $\hat{m}^i_j(t)$), then it would be simulating play of the n player game where each player would be using the "Multiplicative Weights Update Algorithm with Restricted Distributions" from [Arora et al. 2012], where each player is restricted to playing a distribution in $K_{\epsilon/6}$. The guarantee of the multiplicative weights update algorithm with restricted distributions (see e.g. [Arora et al. 2012] Theorem 2.4) gives us (where $m^i(t)$ denotes the loss vector $(m^i_j(t))_j$):

$$\frac{1}{T}\sum_{t=1}^{T} m^i(t)\cdot \mathbf{p}^i(t) \leq \min_{p\in K_{\epsilon/6}} \frac{1}{T}\sum_{t=1}^{T} m^i(t)\cdot p + \eta + \frac{\ln 2}{\eta T}$$

We now make several observations. First, note that for every probability distribution p, there exists a distribution $\hat{p}\in K_{\epsilon/6}$ such that $d_{TV}(p,\hat{p})\leq \epsilon/3$, where d_{TV} represents the total variation distance. Therefore, since the losses $m^i(t)\in[0,1]$ we can deduce:

$$\frac{1}{T}\sum_{t=1}^{T} m^i(t)\cdot \mathbf{p}^i(t) \leq \min_{p} \frac{1}{T}\sum_{t=1}^{T} m^i(t)\cdot p + \eta + \frac{\ln 2}{\eta T} + \epsilon/3$$

which follows from Holder's inequality.

Similarly, as observed by [Kearns et al. 2013], if we have that $\max_{i,j,t}|m^i_j(t)-\hat{m}^i_j(t)|\leq \beta$, then we can deduce:

$$\frac{1}{T}\sum_{t=1}^{T} \hat{m}^i(t)\cdot \mathbf{p}^i(t) \leq \min_{p} \frac{1}{T}\sum_{t=1}^{T} \hat{m}^i(t)\cdot p + \eta + \frac{\ln 2}{\eta T} + \epsilon/3 + 2\beta$$

Proposition 2.3 gave conditions for the observed losses to be within β of true losses.

Applying Proposition 2.3 with $\beta=\epsilon/6$ and $\gamma=\epsilon/6$, we have that with probability at least $1-\delta/T$ the following value of N is sufficient for observed losses to be within β of true losses, at any iteration:

$$N \geq \frac{216}{\epsilon^3}\log\left(\frac{8nT}{\delta}\right). \tag{3}$$

We therefore have:

$$\frac{1}{T}\sum_{t=1}^{T} \hat{m}^i(t)\cdot \mathbf{p}^i(t) \leq \min_{p} \frac{1}{T}\sum_{t=1}^{T} \hat{m}^i(t)\cdot p + \eta + \frac{\ln 2}{\eta T} + 2\epsilon/3$$

APPROX COARSE CE (2)
Parameters: learning rate $\eta = \frac{\epsilon}{6}$; number of iterations $T = \frac{36 \ln 2}{\epsilon^2}$; sample size $N = \frac{216}{\epsilon^3} \log\left(\frac{8nT}{\delta}\right)$.

(a) Let $\mathbf{p}_j^i(t)$ denote the probability mass on strategy $j \in \{0,1\}$ of player i at iteration t; initially $\mathbf{p}_j^i(1) = 1/2$ for all i, $j \in \{0,1\}$.

(b) $K_\epsilon = \{(p_0, p_1) : p_0 + p_1 = 1 \text{ and } \min(p_0, p_1) \geq \epsilon\}$ is the convex body defined to be the 2-dimensional probability simplex intersected with the constraints that p_0, p_1 must both be greater than or equal to ϵ.

Do the following for iterations $t = 1, \ldots, T$:

(1) For each player i let $\hat{\mathbf{p}}^i(t) = \arg\min_{p \in K_{\epsilon/6}} \mathrm{RE}(p || \mathbf{p}^i(t))$. Here RE is the relative entropy function.

(2) Let $\mathbf{p}(t)$ be the product distribution over mixed-strategy profiles that results from each player i playing $\hat{\mathbf{p}}^i(t)$.

(3) Sample N pure-strategy profiles $S(t) = \{\mathsf{s}_1(t), \ldots, \mathsf{s}_N(t)\}$ from $\mathbf{p}(t)$.

(4) query the payoffs for all the profiles in $S(t)$. Let $\hat{m}_j^i(t)$ be the empirical cost of strategy j when used by player i. (That is, letting $S_j^i(t) = \{s \in S : i \text{ plays } j \text{ in } S\}$, $m^i(s)$ be 1 minus the payoff to i for $s \in S_j^i(t)$, then $\hat{m}_j^i(t) = |S_j^i(t)|^{-1} \sum_{s \in S_j^i(t)} m^i(s)$. If i never plays j in the profiles $S(t)$, the algorithm fails.)

(5) update weights: for each i, j, set $\mathbf{p}_j^i(t+1) = \hat{\mathbf{p}}_j^i(t)(1 - \eta \hat{m}_j^i(t))/\phi$ where ϕ is the normalization constant needed to make $\mathbf{p}^i$ a probability distribution.

Output: The distribution $\mathbf{p} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{p}(t)$.

Fig. 2. Using Multiplicative Weights to compute an approximate coarse CE

Plugging in our values for T and η bounds the regret of each player by at most:

$$\frac{1}{T} \sum_{t=1}^{T} \hat{m}^i(t) \cdot \mathbf{p}^i(t) - \min_p \frac{1}{T} \sum_{t=1}^{T} \hat{m}^i(t) \cdot p \leq \epsilon$$

which proves the theorem. □

We believe Theorem 2.4 could generalize to m actions, at the cost of a factor-m increase in sample complexity. We focus on the 2-action case since in this case, an ϵ-approximate coarse correlated equilibrium is a 2ϵ-approximate correlated equilibrium, so we have the following result:

COROLLARY 2.5. *For binary-choice games (where each player has 2 pure strategies), for any ϵ, with probability $1 - \frac{1}{n}$, Algorithm* APPROX COARSE CE (2) *finds an ϵ-CE, using $\tilde{O}(\frac{\log n}{\epsilon^5})$ payoff queries.*

2.3. Lower bound for n players, binary actions

For positive integer $k > 2$ we lower-bound the number of queries needed to find a $(\frac{1}{2} - \frac{1}{k})$-approximate CE, by fixing the number of queries to be $Q = \lfloor \log_{k(k-1)} n \rfloor$ and applying Yao's minimax principle. The cost of the output of algorithm $\mathcal{A}$ is the smallest value of ϵ for which $\mathcal{A}$'s output is an ϵ-approximate CE. We identify a distribution over payoff functions of n-player binary-action games such that the expected cost of the solution output by any algorithm $\mathcal{A}$ that uses Q queries (or fewer) is approximately $\frac{1}{2}$.

A distribution over payoff functions. Define the following probability distribution over payoffs of n-player binary-choice games. Each player $i \in [n]$ shall be a "type-0" player or a "type-1" player, a player's type being obtained by flipping a fair coin. Let t_i be the type of player i. After types have been obtained, for each player i, construct i's payoff function as follows. For each strategy profile s_{-i} of players other than i, with probability $\frac{k-1}{k}$ player i gets paid 1 to play t_i and 0 to play $1 - \mathsf{t}_i$. With probability $\frac{1}{k}$, i gets paid 0 to play t_i and 1 to play $1 - \mathsf{t}_i$. Hence, for a (expected) fraction $\frac{k-1}{k}$ of profiles s_{-i}, player i has t_i as best response, with $1 - \mathsf{t}_i$ as the best response for the remaining profiles s_{-i}.

Notation. Let $\mathcal{A}$ be an algorithm that makes Q pure profile queries, for $Q = \log_{k(k-1)} n$. Let $\mathsf{s}^1, \ldots, \mathsf{s}^Q$ be the queried profiles, where s^j is the j-th query in the sequence made by $\mathcal{A}$.

An important observation is that, conditional on a choice of player types, all payoff vectors are generated independently of each other. Consider the process of generating the payoff function as above, and then querying it. That process is equivalent to one where the type vector is initially generated, then the algorithm selects various pure-strategy profiles to query, and then each time a pure-strategy profile is queried, we flip the biased coins that produce its payoff vector. This shows a useful limitation on how the answers to a sequence of queries can indicate what the answer will be to any subsequent query.

PROPOSITION 2.6. *For $0 \leq j \leq Q$, let p^j be the probability distribution over type vectors, conditioned on the answers to the first j queries (i.e. the players' payoffs for $\{\mathsf{s}^1, \ldots, \mathsf{s}^j\}$). Then p^j is a product distribution, $\mathsf{p}^j = \times_i \mathsf{p}_i^j$, where p_i^j is the probability that player i has type 1.*

PROOF. The claim follows by induction on j. Initially, p^0 is the uniform distribution. Subsequently, the payoffs for each queried profile s^j are obtained by making a (biased) coin-flip independently for each player. (After the type vector has been selected, we can assume that the payoffs for a queried pure profile, are generated by making biased coin-flips, independently of the results of previous queries.)

In particular, when $\mathcal{A}$ queries strategy profile s^j, $\mathcal{A}$ observes a payoff for each player i consisting of the result of a coin-flip which is

— equal to 1 with probability $\frac{k-1}{k}$ and 0 with probability $\frac{1}{k}$ if $\mathsf{t}_i = \mathsf{s}_i^j$, and
— equal to 1 with probability $\frac{1}{k}$ and 0 with probability $\frac{k-1}{k}$ if $\mathsf{t}_i = 1 - \mathsf{s}_i^j$.

□

OBSERVATION 3. *We say that the payoffs for s^j indicate that player i is type $t \in \{0, 1\}$ if i plays 1 and gets paid t, or i plays 0 and gets paid $1-t$. Let Q_i^i be the number of queries that indicate that player i is type t. It is not hard to check that $\Pr[\mathsf{t}_i = 1]/\Pr[\mathsf{t}_i = 0] = (k-1)^{(2Q_1^i - Q)}$, where $\Pr[\mathsf{t}_i = t]$ is the probability that i has type t, conditioned on the data.*

Definition 2.7. $\mathcal{A}$ outputs a distribution ψ over pure-strategy profiles. Let ψ^u be ψ restricted to the un-queried profiles. We will say that $\mathcal{A}$ *has bias b for player i*, if on profiles sampled from ψ^u, i plays 1 with probability b, i.e. $\mathbb{E}_{x \sim \psi^u}[x_i] = b$ where x_i is the action played by i in profile x.

THEOREM 2.8. *Let $k > 2$ be an integer. Let $\mathcal{A}$ be a payoff-query algorithm that uses at most $\log_{k(k-1)} n$ queries, where n is the number of players, and outputs distribution ψ.*

With probability more than $\frac{1}{2}$ there will exist a player i who would improve his payoff by $\frac{1}{2} - \frac{1}{k}$ by playing some fixed strategy in $\{0,1\}$, relative to the payoff he gets in ψ.

PROOF. Assume that the payoffs for game G are generated by the distribution defined above.

We identify a lower bound on the probability that any individual player i is paid 0 in every profile in $\{\mathsf{s}^1, \ldots, \mathsf{s}^Q\}$, and has a type t_i that is in a sense "bad" for ψ.

We start by lower-bounding the probability that a given player gets paid zero on every query. To this end, suppose that $\mathcal{A}$ tries to maximize the probability that each player gets paid at least 1. This is done by selecting s_j that allocates to each player i, the action that is more likely to pay 1 to i, conditioned on the answers to $\mathsf{s}_1, \ldots, \mathsf{s}_{j-1}$. However, since i's payoff is obtained by a coin flip that pays i zero with probability either $\frac{1}{k}$ or $\frac{k-1}{k}$, at each query there is probability at least $\frac{1}{k}$ that i will get paid 0. Hence with probability at least $\frac{1}{k}^Q$, i is paid zero on all queries.

Conditioned on the answers to all Q queries, we have (noting Observation 3) for any player i that $\Pr[\mathsf{t}_i = 1]/\Pr[\mathsf{t}_i = 0] \leq (k-1)^Q$ and similarly $\Pr[\mathsf{t}_i = 0]/\Pr[\mathsf{t}_i = 1] \leq (k-1)^Q$, where $\Pr[\mathsf{t}_i = t]$ is the conditional probability that t_i is equal to t. Hence, any prediction of a player's type has probability at least $(k-1)^{-Q}$ of being incorrect, regardless of how much that player was paid during the queries.

Consider some player i who gets paid zero on all Q queried profiles. Let b be the bias (Definition 2.7) for player i and let λ be the probability that $x \sim \psi$ happens to be a queried profile. Thus $\psi = \lambda\psi^q + (1-\lambda)\psi^u$ where ψ^q is a distribution over queried profiles and ψ^u is a distribution over unqueried profiles.

Let p_0 (resp. p_1) be the probability that $x \sim \psi$ is a queried profile where i plays 0 (resp. 1). Let p'_0 (resp. p'_1) be the probability that $x \sim \psi$ is an unqueried profile where i plays 0 (resp. 1). Thus $p_0 + p_1 + p'_0 + p'_1 = 1$; $p_0 + p_1 = \lambda$; $p'_0 = (1-\lambda)(1-b)$; $p'_1 = (1-\lambda)b$.

Suppose player i is paid 0 in all queried profiles.

— If $\mathsf{t}_i = 0$, then i's expected payoff under ψ is $p'_0\frac{k-1}{k} + p'_1\frac{1}{k}$.
— If $\mathsf{t}_i = 1$, then i's expected payoff under ψ is $p'_1\frac{k-1}{k} + p'_0\frac{1}{k}$.

Furthermore:

— If $\mathsf{t}_i = 0$ and i plays 0 against all profiles generated by ψ, i's expected payoff is

$$p_1 + (1-\lambda)\frac{k-1}{k} = p_1 + \frac{k-1}{k}(p'_0 + p'_1).$$

— If $\mathsf{t}_i = 1$ and i plays 1 against all profiles generated by ψ, i's expected payoff is

$$p_0 + (1-\lambda)\frac{k-1}{k} = p_0 + \frac{k-1}{k}(p'_0 + p'_1).$$

Suppose $\mathsf{t}_i = 0$. i can improve his expected payoff by $p_1 + \frac{k-1}{k}(p'_0+p'_1) - (p'_0\frac{k-1}{k} + p'_1\frac{1}{k})$ by playing 0 always. Suppose $\mathsf{t}_i = 1$. i can improve his expected payoff by $p_0 + \frac{k-1}{k}(p'_0 + p'_1) - (p'_1\frac{k-1}{k} + p'_0\frac{1}{k})$ by playing 1 always. So, there exists a value for t_i such that i's regret is at least

$$\max\Big\{p_1 + \frac{k-1}{k}(p'_0+p'_1) - \Big(p'_0\frac{k-1}{k} + p'_1\frac{1}{k}\Big),\ p_0 + \frac{k-1}{k}(p'_0+p'_1) - \Big(p'_1\frac{k-1}{k} + p'_0\frac{1}{k}\Big)\Big\}$$

which simplifies to

$$\max\Big\{p_1 + \frac{k-2}{k}p'_1,\ p_0 + \frac{k-2}{k}p'_0\Big\}.$$

The sum of these two terms is $p_0 + p_1 + \frac{k-2}{k}(p'_0 + p'_1)$, and since $p_0 + p_1 + p'_0 + p'_1 = 1$, the sum of the terms is at least $\frac{k-2}{k}$, so at least one of the terms is at least $\frac{k-2}{2k}$, hence the maximum of them is at least $\frac{k-2}{2k}$.

For a player i to have regret at least $\frac{k-2}{2k}$ it is sufficient for the following 2 events to occur: player i is paid 0 on all queried profiles, and player i turns out to have type t_i that leads to larger regret than the alternative type $1 - \mathsf{t}_i$. The first of these events occurs with probability at least $\frac{1}{k}^Q$, and (given the first) the second occurs with probability at least $(\frac{1}{k-1})^Q$. So for any player i, with probability at least $(\frac{1}{k(k-1)})^Q$, i has regret at least $\frac{k-2}{2k}$. Thus for $n \geq (k(k-1))^Q$, i.e. $Q \leq \log_{k(k-1)} n$, a bad player exists with probability more than $\frac{1}{2}$. □

3. BOUNDS FOR THE QUERY COMPLEXITY OF ϵ-WSCE

In Section 3.1 we present a lower bound that is linear in n. We then present an upper bound in Section 3.2 that applies to the class of concisely-representable games and finds an ϵ-WSNE. In Section 3.3 we show that this upper bound takes the form of a generic reduction from the problem of computing ϵ-WSNE to the problem of verifying ϵ-WSNE, and can be implemented computationally efficiently in certain games.

3.1. Lower bound

To obtain a lower bound that applies to randomized algorithms, in a similar way to Theorem 2.8 we define an adversarial distribution over games in $\mathcal{G}_n$ and argue that given a deterministic algorithm $\mathcal{A}$ that uses $o(n)$ queries, that $\mathcal{A}$ is likely to fail to find an ϵ-WSCE.

A distribution over payoff functions. Let $\mathcal{D}_n$ be the following distribution over $\mathcal{G}_n$. For each player i, and each bit vector $\mathbf{b}$ of length $i-1$, let $b_{i,\mathbf{b}} \in \{0,1\}$ be obtained by flipping a fair coin. If players $1, \ldots, i-1$ play $\mathbf{b}$ then i is paid $b_{i,\mathbf{b}} \in \{0,1\}$ to play 0 and $1 - b_{i,\mathbf{b}} \in \{0,1\}$ to play 1. Note that every game generated by $\mathcal{D}$ has a unique (pure) NE.

THEOREM 3.1. *For $\epsilon < 1$, the payoff query complexity of computing ϵ-WSCE of $\mathcal{G}_n$ (i.e. n-player 2-action games with payoffs in $[0,1]$) is $\Omega(n)$.*

PROOF. For any game G in the support of $\mathcal{D}_n$, observe that G has a unique (pure) Nash equilibrium $\mathcal{N}$. $\mathcal{N}$ is found by considering each player i in ascending order, and noting that each player is incentivized to select one of his actions based on the behavior of $1, \ldots, i-1$, and i's payoffs are not a function of the behavior of $i+1, \ldots, n$. Moreover, $\mathcal{N}$ can be seen to be the unique ϵ-WSCE of G.

For $G \sim \mathcal{D}_n$, let $\mathsf{s}_1, \ldots, \mathsf{s}_j$ be a sequence of query profiles for G and consider the conditional distribution $\mathcal{D}'$ on $\mathcal{G}_n$ that results from the answers to those queries. We claim that $\mathcal{D}'$ has the following form. Let m be the length of the longest prefix of players that all get paid 1 in some query,

$$m = \arg\max_m \{\exists q \in 1 \ldots j \ : \ \mathsf{s}_q \text{ pays 1 to } 1, \ldots, m\}.$$

Let s be the queried profile that pays 1 to players $1, \ldots, m$ and 0 to player $m = 1$. Then $\mathcal{D}'$ is uniform over all games G in the support of $\mathcal{D}_n$ that pay 1 to players $1, \ldots, m$ to play as in s and pay player $m+1$ 0 to play as in s.

The claim can be proved by induction on j. Let $\mathcal{D}^j$ be the conditional distribution that results from the first j queries, and consider query s_{j+1}. Define m as above. $\mathcal{D}^j$ is uniform over elements of $\mathcal{G}_n$ in its support. $\mathcal{D}^{j+1}$ is obtained by taking the uniform distribution over any such game that is consistent with s_{j+1}.

Define the *progress* of query s_{j+1} to be the increase in the length of the prefix of players whose behavior is determined by elements of $\mathcal{D}^j$. Then the expected progress of each query is less than $1 + \sum_{r \geq 0} r/2^{r+1} = 2$. Meanwhile, the total progress of all queries required to find an ϵ-WSCE is n. □

Using the above result we can obtain a separation between the query complexity of computing small support ϵ-CE and arbitrary support ϵ-CE for any $\epsilon \geq (\log(n)/n)^{1/3}$. Without loss of generality, we can view approximate correlated equilibria as uniform distributions over (multi)sets of action profiles S: note that multiplicative weights explicitly finds correlated equilibria of this form, and any correlated equilibrium can be put in this form with arbitrarily small loss in the approximation factor by sampling. We note that multiplicative weights finds a correlated equilibrium with a support size $|S|$ that depends on n: $|S| \geq \log(n)/\epsilon^2$. We show that no algorithm with polylogarithmic query complexity can find an approximate correlated equilibrium with support size $O(1/\epsilon)$ (independent of n). Of course such a separation is vacuous in games in which there are no ϵ-CE with support $O(1/\epsilon)$, but in any game that has a pure strategy (ϵ)-Nash equilibrium, there is always a small support CE – in particular, one with support just 1. On this topic, [Babichenko et al. 2013] show that multiplayer games have approximate CEs with polylogarithmic support size, and approximate CCEs with logarithmic support size. They note as an open problem the existence of CEs whose support size depends only on ϵ and not n.

COROLLARY 3.2. *The query complexity of computing ϵ-CE with support size $|S| \leq 1/\epsilon$ is $\Omega(n)$: strictly greater than the query complexity of computing ϵ-CE with support size $|S| > \log(n)/\epsilon^2$ for any constant ϵ.*

PROOF. If ψ is an ϵ-CE that is supported over $|S|$ strategy profiles a, then ψ is also a $|S|.\epsilon$-WSCE. This is because the probability p of any action a with $a^i = j$ with positive probability in ψ must be at least $1/|S|$.

Since Theorem 3.1 gives an $\Omega(n)$ lower bound for computing $\Omega(1)$-WSCE, this in particular also implies an $\Omega(n)$ lower bound for computing a CE with support $O(1/\epsilon)$. This is in contrast to the $O(\log(n)/\epsilon^5)$ upper bound of Theorem 2.4 for computing CE with larger support (in particular, support $\log(n)/\epsilon^2$). □

3.2. Upper bound

We give an upper bound on the query complexity of ϵ-WSNE (and hence ϵ-WSCE) that is polynomial in the number of players n and strategies m, together with the description length of the target game. We leave the query complexity of ϵ-WSCE for unrestricted n-player games (i.e. those that have no polynomial length description) open. Recall that for unrestricted games, the query complexity of ϵ-WSNE is exponential in n [Babichenko 2013] for constant ϵ and m. The class of games used by [Babichenko 2013] are based on random walks on the n-dimensional hypercube; notice that to write down a description of a generic member of this class would require a string of length exponential in n. Thus, any polynomial query algorithm for ϵ-WSCE for general games would have to (unlike our algorithm) avoid also computing ϵ-WSNE. Note that in contrast, Algorithms APPROX COARSE CE and APPROX COARSE CE (2) do not require games to come from a concisely-represented class.

Our upper bound applies just to the query complexity, and it remains an open problem whether a computationally-efficient approach exists in general. It yields a positive result for a special case of the question of query complexity of ϵ-NE using randomized algorithms [Hart and Nisan 2013].

Algorithm APPROX WSNE of Figure 3 can be viewed as a query efficient reduction from the problem of computing an ϵ-CE to the problem of verifying one, using the fol-

APPROX WSNE
Let V be the version space, initially $V = \mathcal{G}_n$, where $|\mathcal{G}_n| \leq 2^{p(n)}$.
Repeat the following until an output is obtained.

(1) Construct game G' as follows. For each profile x, each player i is paid the median payoff that i obtains in x for elements of V;
(2) Compute an $\frac{\epsilon}{2}$-WSNE $\mathcal{N}$ of G';
(3) For every player i, strategy j let $\mathcal{N}_j^i$ be a distribution over strategy profiles obtained by sampling from $\mathcal{N}$ and setting i's strategy to j; let S_j^i be a set of pure profiles sampled u.a.r. from $\mathcal{N}_j^i$, where $|S_j^i| = (\frac{4}{\epsilon}) \log(2p(n))$; query all $x \in S_j^i$;
(4) If any x queried above is inconsistent with G' (in terms of the payoffs resulting from the query) then update V, else halt and output $\mathcal{N}$.

Fig. 3. Query-efficient algorithm for ϵ-WSNE

lowing "halving algorithm" approach. The "concisely representable" constraint means that the number of n-player games is upper-bounded by an exponential function of n. We maintain a "version space", the set of all games that are consistent with queries that have been made so far. If we can find a query that is inconsistent with some constant fraction of the games in the version space, then we reduce the size of the version space by a constant fraction, and hence only polynomially many such queries are sufficient to identify the target game. In each iteration of the algorithm, we construct a proposed solution $\mathcal{N}$ (a probability distribution over strategy profiles) and we sample from it. We query the payoffs associated with each sample. With high probability, if $\mathcal{N}$ is not a valid ϵ-WSNE, it will generate a sample that is inconsistent with at least half the elements of the version space.

THEOREM 3.3. *Let $\mathcal{G}_n$ be a class of n-player, m-strategy games whose elements can be represented using bit strings of length at most $p(n)$. With probability at least $\frac{1}{2}$, Algorithm* APPROX WSNE *(Figure 3) identifies an ϵ-WSNE using $O(nmp(n)(\log p(n))/\epsilon)$ payoff queries.*

PROOF. We prove that at each iteration, with high probability, either a profile is queried that is inconsistent with at least half of the elements of V, or the algorithm halts and outputs an ϵ-WSNE of the target game. Hence the number of iterations is at most $p(n)$.

We consider two cases. Let $G(x)$ denote the payoff vector for game G on profile x. Let G^* be the target game.

Case 1: For all i, j, $\Pr_{x \sim \mathcal{N}_j^i}[G^*(x) \neq G'(x)] < \frac{\epsilon}{4}$.

It follows from the condition of case 1 that for any i, j, the payoff that i gets for j in response to $\mathcal{N}$ in game G^*, is within $\frac{\epsilon}{4}$ of the payoff that i gets for j in response to $\mathcal{N}$ in game G'.

Since $\mathcal{N}$ is an $\frac{\epsilon}{2}$-WSNE of G', if i plays j with positive probability in $\mathcal{N}$, then the payoff i gets for j in game G' in response to $\mathcal{N}$ is at most $\frac{\epsilon}{2}$ less than the payoff i gets for any j' in game G' in response to $\mathcal{N}$.

It follows that if i plays j with positive probability in $\mathcal{N}$, then the payoff that i gets for j in $\mathcal{N}$ in game G^* is at most ϵ less than the payoff that i get for any j' in $\mathcal{N}$ in game G^*. Hence $\mathcal{N}$ is an ϵ-WSNE of G^*, so $\mathcal{N}$ constitutes an acceptable output. There is also a small probability that some x is found for which $G^*(x) \neq G'(x)$. By construction of G', the payoffs resulting from the query of x are inconsistent with at least half of the elements of V.

Case 2: For some i, j, $\Pr_{x \sim \mathcal{N}_j^i}[G^*(x) \neq G'(x)] \geq \frac{\epsilon}{4}$.

In this case it is not hard to check that $|S_j^i|$ is large enough that with probability more than $1 - \frac{1}{2p(n)}$, S_j^i contains a pure profile x whose payoffs under G^* differ from the payoffs under G'. By construction of G', the query of x is inconsistent with at least half of the elements of V.

Since there are at most $p(n)$ iterations, the overall failure probability (an iteration where Case 2 arises and the sample does not contain x for which $G^*(x) \neq G'(x)$) is at most $\frac{1}{2}$. The number of queries at each iteration is $nm(\frac{4}{\epsilon})\log(2p(n))$. □

3.3. A class of efficient algorithms

In this section, we generalize our query-efficient algorithm for finding ϵ-WSNE, and instantiate an efficient version of it for classes of games that have payoff functions with concise linear representations. Consider how our algorithm worked:

(1) We had a mechanism to generate some hypothesis game G' given a collection of play profiles x queried so far.
(2) We compute a WSNE $\mathcal{N}$ of G' and query G to check if $\mathcal{N}$ is an ϵ-WSNE of G. If yes, we output $\mathcal{N}$. If no, we update our hypothesis G' with the new queries we have made, and repeat.

The algorithm works because every time we compute a distribution $\mathcal{N}$ which is a WSNE of G' but not of G, we find a profile x which has payoff differing between G and G' by at least $\epsilon/2$: in other words, a query that witnesses that our algorithm for predicting a hypothesis G' made a significant mistake. Moreover, we have a polynomial upper bound on how many mistakes our prediction algorithm can make. The running time of the algorithm is dominated by two computations: Generating the hypothesis G', and computing the WSNE of G'. In our general reduction from the last section, both are computationally expensive.

This framework suggests a more general approach. We can instantiate it with *any mistake bounded learning algorithm* for player payoff functions in G. Specifically, suppose we have a learning algorithm A that for any sequence of payoff-query/answer pairs $(x_1, a_1), \ldots, (x_m, a_m)$ produces a hypothesis $f = A((x_1, a_1), \ldots, (x_m, a_m))$ which maps play profiles x to payoff values $f(x)$, one for each player. (That is, f represents a hypothesis game G'). Say that the algorithm A makes a mistake with respect to a game G if it errs on its prediction $f(x)$ of the payoff of some queried profile x by more than $\epsilon/2$. Suppose furthermore that for any game G in a restricted class, it is guaranteed that A can never make more than B mistakes. We then have an algorithm for computing ϵ-WSNE using only $B \cdot Q$ queries, where Q is the number of queries needed to check if a distribution $\mathcal{N}$ is an ϵ-WSNE. Moreover, if algorithm A is efficient, and WSNE can be computed efficiently for every hypothesis game G' generated in this manner, then the reduction is computationally efficient.

In Lemma 3.4 below, we identify conditions under which MW can be used to obtain new query bounds for certain classes of games. The approach is motivated by Algorithm APPROX WSNE , in which at each step t, a game G^t is constructed, an equilibrium $\mathcal{N}^t$ is obtained for G^t, and if $\mathcal{N}^t$ does not solve the target game, we find an informative strategy profile, obtained by querying responses to $\mathcal{N}^t$.

We consider games where the payoff function can be expressed as a linear function of a limited number of attributes (or features) of strategy profiles. For example consider congestion games with k facilities. The cost to a player is the sum of the costs of facilities he uses, those costs being determined by the number of users of each facility. Thus we extract the following nk features from a strategy profile: for each facility j and $i \in [n]$, an associated feature is set to 1 if i players use j, otherwise it is set to 0. A

payoff query gives an observation of the input/output behavior of this linear function, which we aim to learn. Notice that if we assumed that facility costs were linear in the number of their users, then only k features would be needed for the payoff function. With quadratic costs we could use $2k$ features (the coefficents of the loads, and the squares of the loads).

The problem of finding an ϵ-NE of target game G^* is reducible to learning the coefficients of payoff function f with accuracy ϵ. (A solution to a game whose payoff function approximates G^* will be an approximate solution to G^*.) As described in the full version, Multiplicative Weights can be used to learn approximations of linear functions via queries, with a mistake bound proportional to the L_1 norm of the target function and logarithmic in its dimensionality. If the target function has a low L_1 norm this leads to a good query bound on the equilibrium learning problem.

Let f^* be the linear function associated with the target game G^* and f corresponds to the game G being tested. Let $\mathcal{N}$ be a pure Nash equilibrium of G. Check whether $\mathcal{N}$ is an ϵ-NE of G^* by querying all pure-strategy deviations of every player. If $\mathcal{N}$ is not an ϵ-NE of G^*, we should be able to find an alternative (pure) strategy profile $\mathcal{N}'$ for which in G^* some player i's payoff is $> \epsilon$ higher in $\mathcal{N}'$ than in $\mathcal{N}$, but in G i's payoff in $\mathcal{N}'$ is at most what it is in $\mathcal{N}$. This means that either $\mathcal{N}$ or $\mathcal{N}'$ corresponds to a point x in feature space where $|f^t(x) - f(x)| > \epsilon$.

LEMMA 3.4. *Suppose a target game G^* belongs to a class $\mathcal{G}$ of potential games where*

(1) *any $G \in \mathcal{G}$ has a payoff function that can be expressed as a linear function $f(x) = \langle x, y \rangle$, where $x \in [0,1]^d$ is a d-dimensional attribute vector of strategy profiles and y specifies G,*
(2) *given any strategy profile, Q queries are sufficient to search for an ϵ-better response.*

Then for target game G^* with payoff function $f^*(x) = \langle x, y^* \rangle$, letting $\|y^*\|$ denote the L_1 norm of y^*, an ϵ-NE of G^* can be found using $O(Q\|y^*\|^2 \log(d)/\epsilon^2)$ queries. Furthermore, if pure ϵ-NE can be computed efficiently we also have that the search is efficient.

PROOF. Consider Algorithm APPROX NE OF POTENTIAL GAMES (Figure 4). Let f^* denote the linear function corresponding to G^*. To find an ϵ-NE it is sufficient to find a game whose payoff function is given by f' that approximates f^* in the sense that $|f'(x) - f^*(x)| \leq \epsilon$ for all $x \in [0,1]^d$. This reduces the problem to approximately learning a linear function.

We learn an approximation to y^* as follows. Let y^1 be the d-dimensional vector $(1/d, \ldots, 1/d)$. Consider the game G^1 whose payoff function is given by $f^1(x) = \langle x, y^1 \rangle$. Compute a pure ϵ-NE $\mathcal{N}^1$ of G^1 and check (using Q queries) whether $\mathcal{N}^1$ is an ϵ-NE of G^*. If it is, we are done, if not we find a value of x such that $|f^1(x) - f^*(x)| > \epsilon$.

Each weights-update operation (incrementing t) is based on having found a point x^t in the domain where $|f^*(x^t) - f^t(x^t)| > \frac{\epsilon}{4}$. (This uses $f^t(\mathcal{N}^t) \geq f^t(x) - \frac{\epsilon}{2}$ and $f^*(\mathcal{N}^t) \leq f^*(x) - \epsilon$, from which it follows that a suitable x^t is found at Step 4.)

Plugging in $\frac{\epsilon}{4}$ as the discrepancy between the value of f^* and the current f^t, into the mistake bound obtained in the full version, we get a mistake bound of $64 \log(d)\|y^*\|^2/\epsilon^2$, which appropriately upper bounds the number of iterations. □

We give an example of a class of games to which the above lemma can be usefully applied. These are network congestion games of the kind studied in [Fearnley et al. 2013], where the costs of edges are unknown increasing functions of the number of players using them, so that these costs need to be learned in order to compute an equilibrium. In the case of n players sharing a directed acyclic graph with m edges, [Fearnley et al. 2013] shows how an exact equilibrium can be found using mn queries.

APPROX NE OF POTENTIAL GAMES
Let game G^1 have payoff function $f^1(x) = \langle x, y^1 \rangle$ where $y^1 \in \mathbb{R}^d_{\geq 0}$ is the initial hypothesis of the multiplicative weights learning algorithm for linear functions. (See the full version for details.)
For $t = 1, 2, \ldots$ do the following until a solution is found.

(1) Compute an $\frac{\epsilon}{2}$-NE $\mathcal{N}^t$ of G^t;
(2) Use Q queries to G^* to check whether a player can improve his payoff by $> \epsilon$ (in G^*);
(3) If not, halt and output $\mathcal{N}^t$;
(4) If yes, let x be the strategy profile resulting from the deviation; if $|f^*(x) - f^t(x)| > \frac{\epsilon}{4}$ let $x^t = x$, else if $|f^*(\mathcal{N}^t) - f^t(\mathcal{N}^t)| > \frac{\epsilon}{4}$ let $x^t = -x$;
(5) Feed x^t as the loss vector to the multiplicative weights algorithm for learning linear functions, and receive the new hypothesis vector y^{t+1}. Let game G^{t+1} have payoff function $f^{t+1}(x) = \langle x, y^{t+1} \rangle$.

Fig. 4. Query-efficient algorithm for ϵ-NE

For approximate NE, we next give a query bound dependent only on ϵ and the number of facilities.

THEOREM 3.5. *Consider n-player congestion games over d facilities, where we think of d as being a constant. Assume that each facility j has an increasing cost function c_j that takes values in $[0, 1]$. The number of queries needed to find ϵ-NE is at most $4.2^{2d}.d^2/\epsilon^2$.*

PROOF. Let $\alpha_{i,j} = c_j(i) - c_j(i-1)$ be the extra cost incurred by raising the load on j from $i-1$ players to i players. Notice that the cost function is linear in the following attributes x_{ij} of a strategy profile: $x_{ij} = 1$ if at least i players use j, otherwise $x_{ij} = 0$. Any observed cost is of the form $\sum_{j \in S} \sum_{i=1}^{N_j} \alpha_{ij} x_{ij}$, where S is the set of facilities used by a player, and N_j is the number of players using facility j. Also, the L_1 norm of any target game cannot exceed d since for any facility j we have $\sum_i \alpha_{ij} \leq 1$.

With regard to the value of Q, given a specific pure profile x (for which we seek a ϵ-better response), there are (at most) 2^{2d} queries that need to be made, since the cost of an alternative strategy for a player will depend on which of the 2^d subsets of facilities he is using in x, and which subset of the d facilities he may move to. Finally, efficient computation of ϵ-NE of these games can be done using ϵ-best response dynamics. □

4. CONCLUSIONS AND FURTHER WORK

Query complexity provides a useful criterion for distinguishing the relative difficulty of alternative solution concepts in game theory. Here, we have used this criterion to formally separate the complexity of finding approximate correlated equilibria and finding approximate well supported correlated equilibria. We have extended the analytical toolkit for query complexity, and have proven a polynomial upper bound for computing well supported Nash equilibria (Theorem 3.3) that contrasts with exponential lower bounds in recent work, provided that the game in question has a concise representation. We applied this idea in Section 3.3 to get a robust and generic method for efficiently finding ϵ-well-supported Nash equilibria of certain congestion games.

Our work leaves open the intriguing question of whether finding well supported correlated equilibria is easier (from a query complexity standpoint) than finding well supported Nash equilibria. Is there a polynomial upper bound for well supported cor-

related equilibria for arbitrary (non-concisely represented) games? Are there efficient game dynamics that converge to such equilibria?

REFERENCES

S. Arora, E. Hazan, and S. Kale. 2012. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. Theory of Computing 8 (May 2012), 121–164.

Y. Babichenko. 2013. Query Complexity of Approximate Nash Equilibria. ArXiv tech rept. *1306.6686* (2013).

Y. Babichenko and S. Barman. 2013. Query complexity of correlated equilibrium. ArXiv tech rept. *1306.2437* (2013).

Y. Babichenko, S. Barman, and R. Peretz. 2013. Small-Support Approximate Correlated Equilibria. ArXiv tech rept. *1308.6025* (2013).

A. Blum, J. Jackson, T. Sandholm, and M. Zinkevich. 2004. Preference Elicitation and Query Learning. *Journal of Machine Learning Research* 5 (Dec. 2004), 649–667.

A. Blum and Y. Mansour. 2007. Learning, Regret Minimization and Equilibria. *In* Algorithmic Game Theory, *Cambridge University Press* (2007).

X. Chen, X. Deng, and S-H. Teng. 2009. Settling the complexity of computing two-player Nash equilibria. Journal of the ACM 56, 3 (2009), 14:1–14:57.

W. Conen and T. Sandholm. 2001. Preference Elicitation in Combinatorial Auctions [Extended Abstract]. In *Proceedings of the ACM Conference on Electronic Commerce*. 256–259.

V. Conitzer. 2009. Eliciting Single-Peaked Preferences Using Comparison Queries. *Journal of Artificial Intelligence Research* 35 (2009), 161–191.

C. Daskalakis, P.W. Goldberg, and C.H. Papadimitriou. 2009. The Complexity of Computing a Nash Equilibrium. SIAM J. Comput. 39, 1 (2009), 195–259.

J. Fearnley, M. Gairing, P.W. Goldberg, and R. Savani. 2013. Learning Equilibria of Games via Payoff Queries. In Procs. of 14th ACM EC. 397–414.

J. Fearnley, P.W Goldberg, R. Savani, and T.B. Sørensen. 2012. Approximate Well-supported Nash Equilibria Below Two-thirds. In Procs of the 5th SAGT *LNCS 7615*. 108–119.

J. Fearnley and R. Savani. 2013. Finding Approximate Nash Equilibria of Bimatrix Games via Payoff Queries. Arxiv rept. `http://arxiv.org/abs/1310.7419` (2013).

Y. Freund and R. Schapire. 1999. Adaptive game playing using multiplicative weights. Games and Economic Behavior 29 (Oct. 1999), 79–103.

P.W. Goldberg and A. Pastink. 2014. On the Communication Complexity of Approximate Nash Equilibria. Games and Economic Behavior 85 (May 2014), 19–31.

A. Gupta, A. Roth, and J. Ullman. 2012. Iterative Constructions and Private Data Release. In *Procs of the 9th TCC*. 339–356.

M. Hardt and G. Rothblum. 2010. A multiplicative weights mechanism for privacy-preserving data analysis. In Procs of the 51st FOCS. 61–70.

S. Hart and Y. Mansour. 2010. How long to equilibrium? The communication complexity of uncoupled equilibrium procedures. Games and Economic Behavior 69, 1 (2010), 107–126.

S. Hart and A. Mas-Colell. 2000. A Simple Adaptive Procedure Leading to Correlated Equilibrium. Econometrica 68, 5 (Sept. 2000), 1127–1150.

S. Hart and N. Nisan. 2013. The Query Complexity of Correlated Equilibria. ArXiv tech rept. *1305.4874* (2013).

M. Kearns, M.M. Pai, A. Roth, and J. Ullman. 2013. Mechanism Design in Large Games: Incentives and Privacy. Arxiv rept. `http://arxiv.org/abs/1207.4084` (2013).

S.C. Kontogiannis and P.G. Spirakis. 2010. Well Supported Approximate Equilibria in Bimatrix Games. Algorithmica 57, 4 (2010), 653–667.

C.H. Papadimitriou. 2005. Computing correlated equilibria in multi-player games. In Procs. of the 37th STOC. *ACM*. 49–56.

A. Roth and T. Roughgarden. 2010. Interactive Privacy via the Median Mechanism. In Procs. of the 42nd STOC. *ACM*. 765–774.

H. Tsaknakis and P.G. Spirakis. 2008. An Optimization Approach for Approximate Nash Equilibria. Internet Mathematics 5, 4 (2008), 365–382.